

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

**AMENDMENTS TO THE CLAIMS**

This Listing of Claims will replace all prior versions, and listings, of claims in this application:

**Listing of Claims:**

1. (Previously Presented) A method for processing an input document encoded in an extensible markup language ("XML"), said method comprising:
  - converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML"), said output document encoded in mXML comprising fewer characters than said input document encoded in XML;
  - processing said output document encoded in mXML;
  - identifying a target to which the processed output document encoded in mXML will be next routed; and
  - determining whether said target is capable of processing documents encoded in mXML;
  - if said target is determined to be capable of processing documents encoded in mXML, transmitting the processed output document encoded in mXML to said target; and
  - if said target is determined to be not capable of processing documents encoded in mXML, converting the processed output document encoded in mXML to an output document encoded in XML, and transmitting said output document encoded in XML to said target.

**Application No. 09/843,198 (Conf. No. 8824)      Docket No. RSW9-2001-0016-US1**  
**Reply to Office Action dated June 1, 2006**

2. (Previously Presented) The method of claim 1, wherein said identifying comprises parsing the processed output document.
3. (Previously Presented) The method of claim 2, wherein said identifying comprises identifying a host name string for routing of the processed output document.
4. (Previously Presented) The method of claim 1, wherein said determining comprises referencing a datastore, said datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.
5. (Previously Presented) The method of claim 1, wherein said converting comprises:
  - creating a document tree representation of the input document;
  - obtaining a node count representing a count of nodes in the document tree representation;
  - writing the node count to an mXML buffer;
  - traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising the steps of:
    - generating a node name;
    - generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;
    - generating a child list specifying index values of zero or more nodes which

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

are children of the node; and

generating a node value specification, which is empty if the node has no value;

generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

appending the data buffer to the mXML buffer to form the output document.

6. (Previously Presented) The method of claim 1, wherein said processing comprises :parsing said output document.

7. (Previously Presented) The method of claim 6, wherein :said parsing of said output document comprises extracting routing data from said output document.

8. (Canceled)

9. (Previously Presented) The method of claim 1, wherein said processing comprises:

processing said output document for content based routing if said target is determined to be capable of processing documents encoded in mXML.

10. (Canceled)

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

11. (Previously Presented) The method of claim 1, further comprising:  
processing the converted output document encoded in XML.

12. (Previously Presented) The method of claim 11, wherein processing the converted output document encoded in XML comprises transmitting the converted output document encoded in XML to said target.

13. (Previously Presented) The method of claim 6, wherein processing said output document encoded in mXML comprises:

parsing a node count representing a count of nodes in the document;

parsing a node specification for each of the nodes, further comprising the

steps of:

parsing a node name;

parsing a child list specifying index values of zero or more  
nodes which are children of the node;

parsing an attribute list specifying zero or more (attribute  
name, attribute value) pair references for attributes of the node; and

parsing a node value specification, which is empty if the  
node has no value; and

parsing a data buffer containing attribute names and attribute values  
referenced from the attribute lists and node values referenced from the node value  
specifications.

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

14. (Previously Presented) A method for processing an input document encoded in a machine-oriented extensible markup language ("mXML") , said method comprising:

determining whether said input document will be next routed to a target which is capable of processing documents encoded in mXML;

converting said input document encoded in mXML to an output document encoded in a extensible markup language ("XML") if said target is determined to be not capable of processing documents encoded in mXML, said output document encoded in XML comprising a non-printing character not included in said input document encoded in mXML; and

processing said output document encoded in XML.

15. (Previously Presented) The method of claim 14, wherein said determining comprises :

Identifying a target to which said input document will be next routed; and

determining whether said target is capable of processing documents encoded in mXML.

16. (Previously Presented) The method of claim 15, wherein determining whether said target is capable of processing documents encoded in mXML comprises referencing a datastore, said datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

17. (Previously Presented) The method of claim 14, wherein said converting comprises:

identifying a document tree representation of the input document;

reading a node count from an mXML buffer;

traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising the steps of:

identifying a node name;

identifying an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

identifying a child list specifying index values of zero or more nodes which are children of the node; and

identifying a node value specification, which is empty if the node has no value;

generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

appending the data buffer to the XML buffer to form the output document.

18. (Previously Presented) The method of claim 14, further comprising :  
processing said input document encoded in mXML.

19. (Previously Presented) The method of claim 18, wherein processing said

**Application No. 09/843,198 (Conf. No. 8824)      Docket No. RSW9-2001-0016-US1**  
**Reply to Office Action dated June 1, 2006**

input document encoded in mXML comprises parsing said input document encoded in mXML.

20. (Previously Presented) The method of claim 14, wherein determining whether said input document will be next routed to a target which is capable of processing documents encoded in mXML comprises transmitting said output documents encoded in XML.

21. (Currently Amended) A computer program product embodied on one or more computer-readable media, the computer program product adapted for processing an input document encoded in an extensible markup language ("XML") and comprising:

computer-readable program code configured to convert said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML");

computer-readable program code configured to process said output document;

computer-readable program code configured to identify a target to which the processed output document will be next routed; and

computer-readable program code configured to determine whether said target is capable of processing documents encoded in mXML, and if so, to initiate conversion of said input document to said output document and transmission of one of said input document and said output document to said target.

22. (Previously Presented) The computer program product of claim 21,

**Application No. 09/843,198 (Conf. No. 8824)      Docket No. RSW9-2001-0016-US1**  
**Reply to Office Action dated June 1, 2006**

wherein said computer-readable program code configured to convert said input document encoded in XML to an output document encoded in mXML comprises:

computer-readable program code configured to create a document tree representation of the input document;

computer-readable program code configured to obtain a node count representing a count of nodes in the document tree representation;

computer-readable program code configured to write the node count to an mXML buffer;

computer-readable program code configured to traverse each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising:

computer-readable program code configured to generate a node name;

computer-readable program code configured to generate an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

computer-readable program code configured to generate a child list specifying index values of zero or more nodes which are children of the node; and

computer-readable program code configured to generate a node value specification, which is empty if the node has no value;

computer-readable program code configured to generate a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

computer-readable program code configured to append the data buffer to the



**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

mXML buffer to form the output document.

23. (Previously Presented) The computer program product of claim 21, wherein said computer-readable program code configured to process said output document comprises computer-readable program code for processing a document encoded in mXML comprising:

computer-readable program code configured to parse the document, further comprising:

computer-readable program code configured to parse a node count representing a count of nodes in the document;

computer-readable program code configured to parse a node specification for each of the nodes, further comprising:

computer-readable program code configured to parse a node name;

computer-readable program code configured to parse a child list specifying index values of zero or more nodes which are children of the node;

computer-readable program code configured to parse an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

computer-readable program code configured to parse a node value specification, which is empty if the node has no value; and

computer-readable program code configured to parse a data buffer containing attribute names and attribute values referenced from the attribute lists and

**Application No. 09/843,198 (Conf. No. 8824)      Docket No. RSW9-2001-0016-US1**  
**Reply to Office Action dated June 1, 2006**

node values referenced from the node value specifications; and

computer-readable program code configured to use the parsed document as input for processing.

24. (Currently Amended) A system for processing an input document encoded in an extensible markup language ("XML"), said system comprising:

means for converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML"), said output document encoded in mXML being compact relative to said input document encoded in XML yet conveying data and structure of said input document;

means for processing said output document encoded in mXML;

means for identifying a target to which the processed output document will be next routed; and

means for determining whether said target is capable of processing documents encoded in mXML; and

means for transmitting one of said input document and said output document to said target.

25. (Previously Presented) The system of claim 24, wherein said means for converting said output document encoded in XML to an output document encoded in mXML comprises:

means for creating a document tree representation of the input document;

means for obtaining a node count representing a count of nodes in the document

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

tree representation;

means for writing the node count to an mXML buffer;

means for traversing each node in the document tree representation and  
generating a corresponding node specification in the mXML buffer, further comprising:

means for generating a node name;

means for generating an attribute list specifying zero or more (attribute  
name, attribute value) pair references for attributes of the node;

means for generating a child list specifying index values of zero or more  
nodes which are children of the node; and

means for generating a node value specification, which is empty if the  
node has no value;

means for generating a data buffer containing attribute names and attribute  
values referenced from the attribute lists and node values referenced from the node  
value specifications; and

means for appending the data buffer to the mXML buffer to form the output  
document.

26. (Original) The system of claim 24, wherein said means for processing  
said output document in mXML comprises:

means for parsing the document, further comprising:

means for parsing a node count representing a count of nodes in the  
document;

means for parsing a node specification for each of the nodes, further

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

comprising:

means for parsing a node name;

means for parsing a child list specifying index values of zero or more nodes which are children of the node;

means for parsing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

means for parsing a node value specification, which is empty if the node has no value; and

means for parsing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

means for using the parsed document as input for the processing.

27. (Currently Amended) A method for processing an input document comprising:

determining whether said input document will be next routed to a target which is capable of processing documents encoded in a machine-oriented extensible markup language ("mXML"); ~~and~~

converting said input document to an output document encoded in an extensible markup language ("XML") if said input document is encoded in mXML and said target is not capable of processing documents encoded in mXML; and

transmitting one of said input document and said output document to said target.

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

28. (Previously Presented) The method of claim 27, further comprising :  
converting an original document encoded in XML to an input document encoded in mXML  
and wherein converting an original document encoded in XML to an input document encoded in mXML occurs prior to determining whether said input document will be next routed to a target which is capable of processing documents encoded in a machine-oriented extensible markup language ("mXML"), said input document encoded in mXML being capable of being processed more efficiently than said input document.

29. (Previously Presented) The method of claim 28, wherein said determining comprises:  
identifying a target to which said input document will be next routed;  
determining whether said target is capable of processing documents encoded in mXML.

30. (Previously Presented) The method of claim 29, wherein said identifying comprises parsing said input document.

31. (Previously Presented) The method of claim 30, wherein said determining comprises referencing a datastore, said datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.

**Application No. 09/843,198 (Conf. No. 8824)  
Reply to Office Action dated June 1, 2006**

**Docket No. RSW9-2001-0016-US1**

32. (Canceled)

33. (Previously Presented) The method of claim 1, wherein said input document encoded in XML comprises a plurality of non-printing characters , and wherein said output document encoded in mXML does not comprise said plurality of non-printing characters .